Augmented Democracy as a Coherence-Constrained Control System

From Oracle Governance to Quantum-Safe Democratic Infrastructure

Sylvain Cormier
Paraxiom Research
sylvain@paraxiom.org

December 2025

Abstract

We present a systems-theoretic framework for democratic governance that treats legitimacy as the preservation of coherence in decision-making processes rather than as a direct consequence of majority outcomes. The framework introduces *epistemic infrastructure*—including token-curated test grids, dynamic credential NFTs, and bounded voting weights—that restrict participation to actors who can demonstrate verifiable engagement with relevant evidence, without assigning semantic authority to the system itself.

Building on prior work in Hamiltonian machine learning (ERLHS), geometric consensus (Karmonic Mesh), and quantum-safe validation (Proof of Coherence), we formalize augmented democracy as a control system in which proposals are state transitions, participants act as distributed sensors with bounded influence, and acceptance requires satisfaction of both a democratic majority condition and a coherence threshold measuring process consistency.

We trace the evolution of this framework from early conceptual work (2017), through EOS-based smart contract implementations, to its current deployment on the Quantum Harmony blockchain using post-quantum cryptography and quantum entropy sources. The central claim is that democratic legitimacy is a measurable property of process quality—independent of specific outcomes—and that governance systems designed around epistemic admissibility and coherence constraints exhibit improved resistance to manipulation, Sybil attacks, plutocratic capture, and coordinated adversarial behavior.

Contents

1 Problem: Democracy Under Adversarial Load

Democratic systems face compounding failure modes in the information age:

1. **Unengaged voting**: Participants lack verifiable engagement with proposal-relevant evidence
2. **Manipulation**: Coordinated influence campaigns exploit cognitive biases
3. **Speed vs. legitimacy**: Rapid decisions sacrifice deliberation quality
4. **AI-generated influence**: Large language models enable scalable persuasion
5. **Quantum-era threats**: Future adversaries with quantum computers can retroactively compromise classical cryptographic commitments

These are not political problems. They are *systems failures*—the democratic control loop lacks adequate filtering, measurement, and stability guarantees.

Traditional responses (education, regulation, fact-checking) address symptoms rather than architecture. We require a formal framework that treats democracy as critical infrastructure subject to engineering constraints.

2 Philosophical Foundations: Artifacts, Not Truth

Before presenting the technical framework, we establish a critical philosophical distinction: **the governance system does not evaluate truth**. It evaluates *procedural admissibility* of evidence artifacts and *process consistency* of decision-making.

2.1 The Semantic Authority Problem

Any system that claims to determine "what is true" faces an infinite regress: who verifies the verifiers? Classical epistemic gatekeeping (expert panels, editorial boards, fact-checkers) ultimately rests on institutional authority that can be captured, corrupted, or contested.

Our framework sidesteps this problem entirely. The system makes no semantic judgments. It enforces *procedural constraints* on what evidence may be referenced and measures *statistical properties* of the decision process.

2.2 Admissible Artifacts

Definition 1 (Admissible Fact Artifact)**.** *In this framework, a* fact *is not treated as a semantic claim or assertion, but as a verifiable artifact with cryptographic provenance. An artifact is admissible for use in governance processes if it satisfies:*

1. ***Immutable referenceability****: the artifact can be uniquely identified via a hash, DOI, or equivalent content-addressed reference;*
2. ***Authentic issuance****: the artifact is bound to a recognized issuer, authority, or origin via digital signatures or equivalent mechanisms;*
3. ***Contextual relevance****: the artifact belongs to an artifact class declared relevant for the proposal type under consideration.*

No semantic interpretation, truth evaluation, or correctness judgment is performed by the governance system itself.

This definition has important consequences:

- A peer-reviewed paper is admissible because it has a DOI, is signed by a journal, and belongs to the "scientific literature" artifact class—not because its conclusions are "true."
- A government report is admissible because it has a document ID, is signed by an agency, and belongs to the "official statistics" artifact class—not because its numbers are "correct."
- Participants demonstrate engagement with admissible artifacts, not agreement with their conclusions.

2.3 Test Grids as Admissibility Registries

Definition 2 (Token-Curated Test Grid)**.** *A Token-Curated Test Grid (TCTG) is a governed registry that specifies which classes of admissible artifacts may be referenced for a given proposal domain. Curation decisions are made by participants who stake tokens and are subject to economic penalties if admitted artifacts are later shown to be malformed, inauthentic, or procedurally invalid.*

Test grids govern admissibility criteria *(format, provenance, relevance), not semantic conclusions or interpretations.*

Critical distinction: Test grids do not define truth; they define the set of artifacts that a decision process is permitted to reference.

Curators are slashed for admitting artifacts that:

- Have invalid provenance (forged signatures, broken hash references)

- Violate format requirements (wrong artifact class for proposal type)
- Were retracted or invalidated by their issuing authority

Curators are *not* slashed for admitting artifacts whose conclusions are later contested, revised, or overturned through normal scientific or institutional processes. The system does not adjudicate semantic disputes.

2.4 Coherence as Process Quality

Definition 3 (Process Coherence)**.** *The coherence score γ does not measure correctness, factual truth, or proposal merit. It measures statistical consistency and entropy quality of the decision process given the declared admissible evidence and participation constraints.*

Low coherence indicates process instability, coordinated behavior, or insufficient entropy, independent of proposal content.

A proposal can have high coherence and still be "wrong" by external standards. A proposal can have low coherence despite being "correct." The coherence score measures whether the *process* that produced the decision exhibited properties consistent with legitimate deliberation.

2.5 Scope of System Authority

The governance system constrains *how* decisions are made, not *what* decisions must conclude.

System Does	**System Does Not**
Verify artifact provenance	Evaluate artifact truth
Enforce participation requirements	Determine correct outcomes
Measure process consistency	Judge proposal merit
Bound individual influence	Override collective judgment
Detect coordinated manipulation	Censor unpopular positions

This scope limitation is not a weakness but a design requirement. A system that claimed semantic authority would be both philosophically indefensible and practically capturable.

3 Augmented Democracy: Definition

3.1 What Augmented Democracy Is Not

- **Not technocracy**: Experts do not override participants
- **Not algorithmic rule**: No AI makes final decisions

- **Not censorship**: All proposals enter the system
- **Not plutocracy**: Wealth does not determine influence
- **Not truth arbitration**: The system does not determine what is "true"

3.2 Definition

Definition 4 (Augmented Democracy)**.** *A **constrained participation system** with admissibility gates, where:*

1. *Participation requires registration (identity binding via dynamic NFTs)*
2. *Voters must demonstrate engagement with proposal-relevant admissible artifacts*
3. *Proposals traverse a filtering pipeline (review period with gatherers/curators)*
4. *Votes are weighted by reputation and bounded by quadratic costs*
5. *Acceptance requires coherence threshold satisfaction (process quality)*
6. *All transitions produce cryptographic proofs (auditability)*
7. *Credentials decay without ongoing participation (life-sustaining NFTs)*

The key distinction from classical democracy: legitimacy derives from *process invariants*, not outcome ratification. The "augmented" refers to procedural augmentation—ensuring voters have verifiably engaged with relevant evidence before their input shapes collective decisions.

4 Procedural Infrastructure: Artifact Curation and Credential Management

The "augmented" in augmented democracy refers not to technological enhancement of voting mechanics, but to *procedural augmentation*—ensuring that participants have verifiably engaged with relevant evidence artifacts before their input shapes collective decisions. This section describes the infrastructure for artifact curation, credential management, and participation constraints.

4.1 The Engagement Problem in Democratic Systems

Classical democratic theory assumes informed voters. Reality provides participants who may not have engaged with relevant evidence. The augmented democracy framework addresses this through *mandatory admissibility gates*—structural requirements that participants demonstrate engagement with proposal-relevant artifacts before voting.

Definition 5 (Admissibility Gate)**.** *A procedural checkpoint* $\mathcal{A} : \mathcal{V} \to \{0, 1\}$ *that maps a potential voter* $v \in \mathcal{V}$ *to eligibility status based on demonstrated engagement with admissible artifacts for the proposal under consideration.*

Critical clarification: This is not a "literacy test" or competency filter. It verifies *engagement with admissible artifacts*, not intelligence, education, political alignment, or agreement with artifact conclusions. A voter who has engaged with the relevant evidence and disagrees with its conclusions still passes the gate.

4.2 Token Curated Test Grids

The admissibility gate mechanism is implemented through **Token Curated Test Grids** (TCTGs), a structure adapted from token curated registries with economic incentives for quality curation.

As established in Section 2, test grids govern *admissibility criteria*—which artifact classes may be referenced for a proposal domain—not semantic conclusions or truth claims.

4.2.1 The Tripartite Structure

Following the KILT Protocol model [8], TCTGs employ three distinct roles:

1. **Claimer**: A participant who asserts readiness to vote on a proposal
2. **Attester**: A curator who has assembled the test grid for that proposal
3. **Verifier**: The system (or designated validators) that checks test completion

This separation prevents conflicts of interest: those who create tests do not administer them, and those who verify do not profit from outcomes.

4.2.2 Gatherers and Curators

Within the Attester role, two sub-functions operate:

- **Gatherers**: Identify admissible artifacts for a proposal—documents with valid provenance from recognized issuers (scientific journals, government agencies, standards bodies). Gatherers are compensated per artifact that meets admissibility criteria.
- **Curators**: Assemble gathered artifacts into test grids that verify engagement. The goal is to confirm that voters have *encountered* the relevant evidence, not that they agree with it. Curators stake tokens on grid quality.

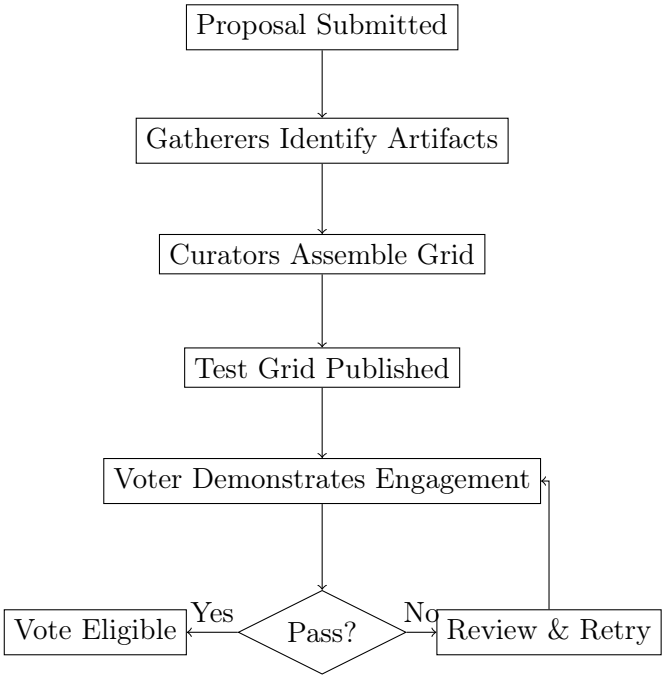


Figure 1: Token Curated Test Grid Flow

4.2.3 Economic Incentives and Slashing Conditions

Token holders curating test grids face a strategic tension:

> "Token holders have a tactical incentive to challenge and reject every candidate to their registry. In the interest of increasing their holdings, this is at odds with their strategic interest of increasing the value of their holdings. An empty list is of no interest to consumers." — KILT Protocol

Applied to TCTGs: curators who create impossible tests drive away voters, reducing the value of the governance token. Curators who create trivial tests undermine procedural quality, also reducing token value. The equilibrium favors *fair tests that accurately verify artifact engagement.*

Slashing conditions (curators lose staked tokens):

- Admitting artifacts with invalid provenance (forged signatures, broken hashes)
- Admitting artifacts from non-recognized issuers
- Admitting artifacts outside the declared artifact class for the proposal domain

- Admitting artifacts that were retracted by their issuing authority

Not slashable (curators are protected):

- Admitting artifacts whose conclusions are later contested or revised
- Admitting artifacts that some participants disagree with
- Admitting artifacts from one scientific position when others exist

The system does not adjudicate semantic disputes. Curators are accountable for *procedural validity*, not *correctness*.

4.3 Dynamic NFTs for Credential Management

Participant credentials in augmented democracy are not static. A voter's eligibility, weight, and privileges evolve based on contribution history. This is implemented through **Dynamic NFTs**—non-fungible tokens whose metadata updates based on on-chain activity.

4.3.1 Credential Lifecycle

Listing 1: Dynamic Credential Structure

```
pub struct DynamicCredential<AccountId> {
    pub holder: AccountId,
    pub credential_type: CredentialType,
    pub issued_at: u64,
    pub last_updated: u64,

    // Dynamic fields (updated on-chain)
    pub reputation_score: u64,
    pub proposals_voted: u32,
    pub engagements_verified: u32,  // Test grids passed
    pub engagements_failed: u32,
    pub contributions: u32,
    pub domains_certified: BoundedVec<DomainId,
        MaxDomains>,

    // Computed eligibility
    pub voting_weight_multiplier: u64,  // basis points
    pub can_submit_proposals: bool,
    pub can_curate_grids: bool,
}

pub enum CredentialType {
    Citizen,        // Basic participation rights
    Contributor,    // Has passed contribution threshold
    Curator,        // Can assemble test grids
```

```
    Validator,        // Can verify consensus
    Guardian,         // Emergency governance rights
}
```

The credential NFT updates automatically when:

- An engagement verification is passed or failed
- A vote is cast
- A proposal is submitted
- Reputation is adjusted by peer review
- Domain certification is earned or revoked

4.3.2 Life-Sustaining NFTs

A critical innovation is the **Life-Sustaining NFT**—a credential that requires ongoing activity to remain valid. Unlike static credentials that persist indefinitely, life-sustaining NFTs decay without continuous participation.

Listing 2: Life-Sustaining Credential Logic

```
pub struct LifeSustainingCredential<BlockNumber> {
    pub base_credential: DynamicCredential,
    pub vitality: u64,              // Current life points
    pub max_vitality: u64,          // Maximum life points
    pub decay_rate: u64,            // Points lost per epoch
    pub last_activity: BlockNumber, // Last qualifying action
    pub revival_cost: Balance,      // Cost to revive if expired
}

impl LifeSustainingCredential {
    pub fn is_alive(&self, current_block: BlockNumber) -> bool {
        let epochs_elapsed = (current_block - self.last_activity)
            / EPOCH_LENGTH;
        let decay = epochs_elapsed * self.decay_rate;
        self.vitality > decay
    }

    pub fn sustain(&mut self, activity_points: u64) {
        self.vitality = min(
            self.vitality + activity_points,
```

```
                self.max_vitality
            );
            self.last_activity = current_block();
        }
}
```

Life-sustaining NFTs address the "ghost voter" problem: credentials issued to participants who subsequently disengage. By requiring periodic activity (voting, contributing, engagement verification), the system ensures that voting weight reflects *active* participation, not historical registration.

4.3.3 Domain-Specific Credentials

Voters may hold credentials in specific domains:

Domain	Unlocks Voting On
Environmental	Climate, conservation, pollution proposals
Technical	Infrastructure, protocol upgrades
Economic	Treasury, tokenomics, funding proposals
Social	Community guidelines, dispute resolution
Emergency	Crisis response, security incidents

Domain credentials are earned by passing domain-specific engagement verifications and maintained through ongoing participation in that domain. A participant may hold multiple domain credentials, each with independent vitality.

4.4 Quadratic Voting for Bounded Influence

The augmented democracy framework incorporates **quadratic voting** to prevent plutocratic capture while preserving signal strength for high-conviction preferences.

4.4.1 The Quadratic Cost Function

The cost to cast n votes on a single proposal from a single participant:

$$\text{cost}(n) = n^2 \tag{1}$$

Votes Cast	Cost	Marginal Cost
1	1	1
2	4	3
3	9	5
4	16	7
5	25	9

The increasing marginal cost discourages concentration of voting power on single proposals, encouraging participants to distribute influence across multiple issues.

4.4.2 Integration with Reputation Weighting

Quadratic voting combines with reputation-based weighting:

$$w_{\text{effective}} = \sqrt{\text{votes\_purchased}} \times r_i \times (1 + \epsilon_i) \tag{2}$$

where r_i is reputation score and ϵ_i is the quantum entropy adjustment from Section 5. The square root of purchased votes ensures diminishing returns, while reputation and entropy preserve the coherence mechanisms.

4.4.3 Whale Resistance

Consider an adversary with 100× the resources of an average participant:

System	Adversary Influence	Ratio
1-person-1-vote	1 vote	1:1
Plutocratic (1:1 stake)	100 votes	100:1
Quadratic	10 votes	10:1
Quadratic + Reputation Cap	$\leq$ 10 votes	$\leq$ 10:1

Quadratic voting reduces the 100:1 wealth advantage to a 10:1 voting advantage. Combined with reputation caps and coherence thresholds, adversarial influence is further bounded.

4.5 Deliberative Democracy: Multi-Round Consensus

For high-stakes proposals, the framework supports **deliberative democracy** through multiple voting rounds with increasing consensus requirements.

4.5.1 Unanimity-Seeking Processes

Drawing from North American Indigenous governance traditions:

> "Unanimity requires that everyone involved agrees."

While perfect unanimity is impractical at scale, the framework supports *unanimity-seeking* processes:

1. **Round 1**: Simple majority required
2. **Round 2**: If Round 1 passes but dissent exceeds threshold, deliberation period opens; 60% supermajority required

3. **Round 3**: If significant dissent remains, face-to-face (or synchronous digital) deliberation; 75% required
4. **Final Round**: Consensus conference with all registered dissenters; 90% required or proposal modified

This process is expensive and slow—by design. It applies only to proposals flagged as "constitutional" or "irreversible."

4.5.2 Dissent Visibility

Unlike anonymous voting, deliberative rounds make dissent *visible* (though not punitive):

> "Shining the light on someone's disagreement within the consensus could help the individual and the collective."

Visible dissent enables:

- Identification of unaddressed concerns
- Opportunity for proposal modification
- Record of minority positions for future reference
- Accountability for officials who override consensus

4.6 Participation History and Accountability

The system maintains participation history for transparency and accountability, without claiming to evaluate correctness.

4.6.1 What Is Recorded

For each participant:

- Engagement verifications passed/failed (procedural record)
- Votes cast and their weights (participation record)
- Proposals submitted and their outcomes (contribution record)
- Coherence scores of votes participated in (process quality record)

4.6.2 What Is Not Recorded or Evaluated

The system does *not*:

- Label votes as "correct" or "incorrect"
- Penalize voters for positions that differ from artifact conclusions
- Evaluate whether voters "should have" voted differently
- Assign semantic meaning to voting patterns

Participants are free to engage with evidence and reach their own conclusions. The system verifies engagement, not agreement.

4.6.3 Official Accountability

For elected officials, participation history is public record:

- Which engagement verifications they passed before voting
- How they voted on each proposal
- Whether they used override authority
- The coherence scores of decisions they participated in

This enables informed electoral choices without the system claiming to evaluate whether official decisions were "correct."

4.7 Node Reputation and Performance History

The procedural infrastructure includes reputation tracking for *infrastructure nodes*, not just human participants.

4.7.1 Oracle Node Reputation

Nodes providing artifact references (document hashes, provenance data) accumulate reputation based on:

- Availability: Uptime and response latency
- Consistency: Variance in reported references
- Validity: Proportion of references that pass provenance checks
- Longevity: Duration of reliable service

This reputation feeds into test grid assembly: artifacts sourced through high-reputation nodes have verified provenance chains.

4.7.2 Validator Performance

Consensus validators are tracked on:

- Block production rate
- Missed slots
- Equivocation incidents
- Coherence score contributions

Poor-performing validators see reduced block rewards and eventual removal from the active set.

4.8 Evolution: EOS to Substrate to Quantum Harmony

The procedural infrastructure evolved through three phases:

4.8.1 Phase 1: EOS Smart Contract Prototype (2020–2021)

Initial implementation as an EOS smart contract:

- Basic test grid structure
- Token-curated artifact registration
- Simple pass/fail admissibility gates
- Proof-of-concept dynamic credentials

4.8.2 Phase 2: Substrate Migration (2022–2023)

Migration to Substrate framework:

- Full pallet implementation
- NFT-based credential system
- Quadratic voting integration
- Multi-round deliberation support

4.8.3 Phase 3: Quantum Harmony Integration (2024–2025)

Current implementation with quantum enhancements:

- Quantum entropy for vote weighting (Section 5)
- Post-quantum signatures for credential integrity
- Coherence-based consensus replacing stake-based finality
- QKD-protected artifact distribution

The core procedural architecture—test grids, dynamic credentials, quadratic voting—persists across all phases. The quantum enhancements provide cryptographic hardening without altering the democratic logic.

4.9 Summary: The Procedural Stack

The complete procedural infrastructure:

1. **Artifact Layer**: Gatherers identify admissible artifacts, nodes verify provenance
2. **Test Layer**: Curators assemble grids, token economics enforce quality
3. **Credential Layer**: Dynamic NFTs track eligibility, life-sustaining requirements enforce engagement
4. **Voting Layer**: Quadratic costs bound influence, reputation weights signal, entropy randomizes
5. **Consensus Layer**: Coherence thresholds measure process quality, multi-round processes seek unanimity
6. **Accountability Layer**: Participation recorded, officials tracked, history immutable

This stack transforms voting from opinion aggregation into *structured preference revelation*—the foundation of augmented democracy.

The system constrains how decisions are made, not what decisions must conclude.

5 Governance as a Control System

This section presents a rigorous mapping from classical control theory to democratic governance, grounded in a production implementation deployed on the Quantum Harmony blockchain. We demonstrate that democratic processes can be formally modeled as coherence-constrained state machines, where proposals represent state transitions, participants act as distributed sensors, and legitimacy emerges from invariant preservation rather than mere majority arithmetic.

5.1 The Control-Theoretic Frame

We model a democratic system as a discrete-time control system $\mathcal{G} = (\mathcal{S}, \mathcal{U}, \mathcal{Y}, f, g, \mathcal{C})$ where:

- $\mathcal{S}$ is the state space (governance configuration)
- $\mathcal{U}$ is the input space (proposals)
- $\mathcal{Y}$ is the output space (decisions)
- $f : \mathcal{S} \times \mathcal{U} \to \mathcal{S}$ is the state transition function
- $g : \mathcal{S} \to \mathcal{Y}$ is the output function
- $\mathcal{C} \subset \mathcal{S}$ is the coherence manifold—the set of legitimate states

The central requirement is *coherence preservation*: for any valid transition $s_{t+1} = f(s_t, u_t)$, if $s_t \in \mathcal{C}$, then $s_{t+1} \in \mathcal{C}$. Transitions that would exit the coherence manifold are rejected.

5.2 Proposals as State Transitions

In the implementation, proposals are formalized as typed state transition requests with explicit lifecycle semantics:

Listing 3: Proposal Status State Machine (lib.rs:114-124)

```
pub enum ProposalStatus {
    Submitted,          // s_0: Initial state
    UnderReview,        // s_1: Filtering phase
    ReadyForVoting,     // s_2: Cleared for deliberation
    VotingActive,       // s_3: Consensus formation
    VotingClosed,       // s_4: Measurement complete
    Approved,           // s_5+: On-manifold transition
        accepted
    Rejected,           // s_5-: Off-manifold transition
        blocked
    QuantumVerified,    // s_6: Cryptographic proof
        generated
}
```

This eight-state machine enforces a strict ordering: $s_0 \to s_1 \to s_2 \to s_3 \to s_4 \to \{s_5^+, s_5^-\} \to s_6$. The system prohibits state regression and enforces temporal guards:

Listing 4: Temporal Constraints on Voting Phase (lib.rs:430-435)

```
let current_block = <frame_system::Pallet<T>>::
    block_number();
ensure!(
```

```
    current_block >= proposal.voting_starts &&
    current_block <= proposal.voting_ends,
    Error::<T>::ProposalNotInVotingPhase
);
```

The `ReviewPeriod` and `VotingPeriod` configuration parameters define minimum dwell times in each phase, preventing rushed transitions that could destabilize the system.

5.3 Participants as Distributed Sensors

Democratic participants are modeled as heterogeneous sensors with role-specific measurement capabilities:

Listing 5: Participant Roles as Sensor Types (lib.rs:88-94)

```
pub enum ParticipantRole {
    Submitter,    // Proposal origination sensor
    Voter,        // Binary/ternary measurement sensor
    Reviewer,     // Quality filtering sensor
    Validator,    // Consensus verification sensor
}
```

Each participant maintains state that influences their measurement weight:

Listing 6: Participant State Vector (lib.rs:77-86)

```
pub struct Participant<AccountId> {
    pub account: AccountId,
    pub role: ParticipantRole,
    pub registered_at: u64,
    pub contributions: u64,
    pub reputation: u64,            // Base control gain
    pub quantum_identity: Option<H256>,
}
```

The `reputation` field serves as the base *control gain*—the amplification factor that determines how strongly a participant's measurement influences the system output. Initial reputation is set to 100 (lib.rs:336), establishing a baseline that can evolve through contribution history.

5.4 Deliberation as Filtering

The review period implements a low-pass filter on the proposal stream:

Listing 7: Review Period as Temporal Filter (lib.rs:392-393)

```
voting_starts: <frame_system::Pallet<T>>::block_number()
    + T::ReviewPeriod::get(),
voting_ends: <frame_system::Pallet<T>>::block_number()
    + T::ReviewPeriod::get() + T::VotingPeriod::get(),
```

This enforced delay serves multiple control-theoretic purposes:

1. **Noise rejection**: Transient proposals (spam, emotional reactions) decay during the review period
2. **Signal integration**: Reviewers accumulate information about proposal quality
3. **Aliasing prevention**: The minimum period ensures adequate sampling of community response

The system implements bounded queuing to prevent denial-of-service:

Listing 8: Bounded Proposal Queue (lib.rs:366-370)

```
let active_proposals = <Proposals<T>>::iter().count() as
    u32;
ensure!(
    active_proposals < T::MaxProposals::get(),
    Error::<T>::MaxProposalsReached
);
```

5.5 Credentials as Control Gains

Vote weight is computed as a function of reputation modulated by quantum entropy:

Listing 9: Control Gain Computation (lib.rs:617-631)

```
fn calculate_quantum_vote_weight(
    participant: &Participant<T::AccountId>,
    entropy: &[u8],
) -> Result<u64, Error<T>> {
    // Base weight from reputation
    let base_weight = participant.reputation;

    // Add quantum randomness factor (+/- 10%)
    let entropy_factor = (entropy[0] as u64 % 21) as i64
        - 10;
    let quantum_adjustment =
        (base_weight as i64 * entropy_factor) / 100;

    let final_weight =
        (base_weight as i64 + quantum_adjustment).max(1)
            as u64;

    Ok(final_weight)
}
```

This formulation has precise control-theoretic semantics:

$$w_i = \max\left(r_i + \frac{r_i \cdot (\eta_i \mod 21 - 10)}{100}, 1\right) \tag{3}$$

where w_i is the final weight, r_i is the base reputation, and η_i is the quantum entropy byte. The entropy introduces a $\pm 10\%$ perturbation, which serves two purposes:

1. **Sybil resistance**: An attacker controlling multiple identities cannot precisely predict aggregate weight
2. **Tie-breaking**: Near-equal coalitions are resolved by entropy rather than timestamp or insertion order

The `max(..., 1)` floor ensures no participant has zero influence—a stability constraint preventing degenerate control configurations.

5.6 Coherence as System Invariant

The central contribution is the dual-condition consensus requirement:

Listing 10: Coherence-Constrained Consensus (lib.rs:666-667)

```
// Proposal approved if majority approves AND
// quantum confidence is high
let approved = approve_weight > reject_weight
    && quantum_confidence > 50;
```

This is not mere majority rule. Approval requires *both*:

1. $w_{\text{approve}} > w_{\text{reject}}$ (democratic condition)
2. $\gamma > 50$ (coherence condition)

where γ is the *quantum confidence score*, computed as:

Listing 11: Quantum Confidence Computation (lib.rs:679-715)

```
fn calculate_quantum_confidence(
    votes: &[QuantumVote<T::AccountId>],
) -> Result<u32, Error<T>> {
    // ... compute mean entropy across votes ...

    for vote in votes {
        let entropy_value = vote.quantum_nonce.iter()
            .map(|&b| b as f64)
            .sum::<f64>() / vote.quantum_nonce.len() as
               f64;

        entropy_variance +=
```

```
            (entropy_value - mean_entropy).powi(2);
    }

    entropy_variance /= votes.len() as f64;

    // Higher variance indicates better quantum
       randomness
    let confidence =
        ((entropy_variance / 255.0).min(1.0) * 100.0) as
           u32;

    Ok(confidence)
}
```

The coherence score measures the *quality of the randomness distribution* across votes:

$$\gamma = \min\left(\frac{\sigma_\eta^2}{255}, 1\right) \times 100 \tag{4}$$

where σ_η^2 is the variance of mean entropy values across all votes. This captures a subtle but critical property: a legitimate vote should exhibit high entropy variance (true quantum randomness), while a coordinated attack (replay, Sybil) will show artificially low variance.

Critical clarification: The coherence score γ measures *process quality*, not outcome correctness. A proposal can have high coherence and still be "wrong" by external standards. A proposal can have low coherence despite being "correct." The score measures whether the *process* exhibited properties consistent with legitimate deliberation—statistical independence of participants and adequate entropy quality—independent of proposal content.

Key insight: The coherence threshold acts as a Lyapunov function. States with $\gamma \leq 50$ are *outside the coherence manifold*—they may satisfy the democratic condition but fail the process quality invariant. The system refuses to transition to such states, even under majority pressure.

5.7 Rejection of Off-Manifold Transitions

The error handling system explicitly enumerates forbidden transitions:

Listing 12: Off-Manifold Action Rejection (lib.rs:273-310)

```
pub enum Error<T> {
    ParticipantNotRegistered,     // Unidentified sensor
    ParticipantAlreadyRegistered,
    ProposalNotFound,             // Invalid reference
    ProposalNotInVotingPhase,     // Temporal violation
    AlreadyVoted,                 // Replay attempt
```

```
    InsufficientQuantumEntropy,   // Entropy pool exhausted
    InvalidQuantumSignature,      // Authentication failure
    Unauthorized,                 // Role violation
    MaxProposalsReached,          // Queue overflow
    MaxVotersReached,             // Capacity limit
    InvalidProposalData,          // Malformed input
    QuantumConsensusFailed,       // Coherence violation
}
```

Each error type corresponds to a specific invariant violation:

Error	Violated Invariant
`InsufficientQuantumEntropy`	Resource availability
`ProposalNotInVotingPhase`	Temporal ordering
`AlreadyVoted`	Idempotency
`Unauthorized`	Role-based access
`QuantumConsensusFailed`	Coherence threshold

5.8 Bounded Influence and Stability

The system enforces hard bounds on participation:

Listing 13: Configuration Bounds (lib.rs:46-56)

```
#[pallet::constant]
type MaxProposals: Get<u32>;

#[pallet::constant]
type MaxVoters: Get<u32>;

#[pallet::constant]
type MaxReviewers: Get<u32>;
```

These bounds serve as *stability constraints*:

- **MaxProposals**: Prevents unbounded growth of the pending state space
- **MaxVoters**: Limits consensus computation complexity to $O(n)$
- **MinQuantumEntropy**: Ensures sufficient randomness reservoir

The entropy pool management (lib.rs:557-574) implements a FIFO queue with bounded capacity:

Listing 14: Entropy Pool Constraint (lib.rs:562-565)

```
ensure!(
    pool_size >= bytes &&
    pool_size >= T::MinQuantumEntropy::get(),
    Error::<T>::InsufficientQuantumEntropy
);
```

Operations fail gracefully when entropy is exhausted, preventing the system from entering a low-coherence regime where randomness quality degrades.

5.9 Cryptographic Proof Generation

Each consensus produces a verifiable proof:

Listing 15: Consensus Proof Generation (lib.rs:718-741)

```
fn generate_quantum_consensus_proof(
    votes: &[QuantumVote<T::AccountId>],
    result: &QuantumConsensusResult,
) -> Result<BoundedVec<u8, ConstU32<1024>>, Error<T>> {
    let mut proof_data = Vec::new();

    // Aggregate vote signatures
    for vote in votes {
        proof_data.extend_from_slice(&vote.
            quantum_signature);
    }

    // Bind to result
    proof_data.extend_from_slice(&result.encode());

    // Add fresh entropy
    let quantum_entropy = Self::consume_quantum_entropy
        (64)?;
    proof_data.extend_from_slice(&quantum_entropy);

    // Commit
    let proof = BlakeTwo256::hash(&proof_data);

    Ok(proof.as_bytes().to_vec().try_into()?)
}
```

This proof commits to:

1. All individual vote signatures (participation record)
2. The computed result (outcome binding)
3. Fresh quantum entropy (temporal uniqueness)

The resulting Blake2-256 hash provides a succinct, collision-resistant commitment that can be verified without reconstructing the full vote set.

5.10 Summary: The Control Loop

The complete governance control loop is:

1. **Input**: Proposal u_t submitted by registered Submitter
2. **Filtering**: Review period applies temporal low-pass filter
3. **Measurement**: Voters provide weighted observations $\{(v_i, w_i)\}$
4. **Aggregation**: Compute $w_{\text{approve}}, w_{\text{reject}}, w_{\text{abstain}}$
5. **Coherence Check**: Compute γ from entropy distribution
6. **Decision**: Accept transition iff $w_{\text{approve}} > w_{\text{reject}} \wedge \gamma > 50$
7. **Proof**: Generate cryptographic commitment to decision
8. **State Update**: Transition to s_{t+1} or reject and remain at s_t

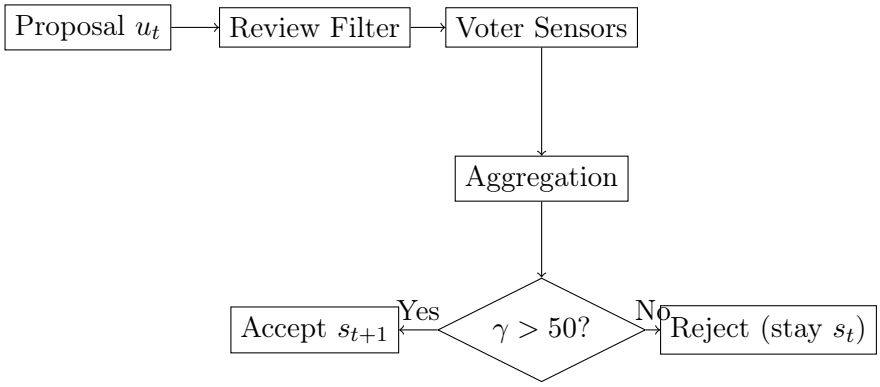


Figure 2: Governance Control Loop with Coherence Gate

The critical observation is that the coherence gate ($\gamma > 50$) is *not optional*. Even unanimous approval fails if the entropy distribution indicates manipulation. This inverts the traditional democratic assumption: legitimacy is not derived from majority agreement, but from the quality of the process that produced the agreement.

5.11 Implications for Democratic Theory

This control-theoretic framing has several implications:

1. **Legitimacy is measurable**: The coherence score γ provides a quantitative measure of process quality, independent of outcome.
2. **Manipulation is detectable**: Coordinated attacks produce low-variance entropy distributions, triggering coherence failure before state corruption.
3. **Stability is guaranteed**: Bounded queues, temporal guards, and entropy floors prevent the system from entering degenerate configurations.
4. **Rollback is possible**: Rejected transitions leave the system at s_t, enabling retry with improved proposals rather than corrupted state recovery.

The implementation demonstrates that coherence-constrained democratic systems are not merely theoretical constructs but deployable infrastructure with formal guarantees.

5.12 Scope of Control

A final clarification on system scope:

The governance system constrains how decisions are made, not what decisions must conclude.

The control system ensures:

- Participants have engaged with admissible artifacts (procedural gate)
- Influence is bounded and distributed (quadratic costs, reputation caps)
- The decision process exhibits statistical independence (coherence threshold)
- All transitions are auditable (cryptographic proofs)

The control system does *not* ensure:

- Outcomes are "correct" by any external standard
- Participants agree with artifact conclusions
- Proposals are wise, beneficial, or optimal
- The collective makes "good" decisions

This scope limitation is deliberate. A system that claimed to guarantee correct outcomes would require semantic authority—the ability to evaluate truth—which is both philosophically indefensible and practically capturable. By limiting scope to process quality, the system remains defensible as infrastructure rather than oracle.

6 The Coherence Pipeline: From Question to Result

The preceding sections describe components. This section shows how they integrate into a single coherent pipeline. The augmented democracy system is not a collection of mechanisms but a *processing pipeline* where each stage gates the next.

6.1 The Six-Stage Pipeline

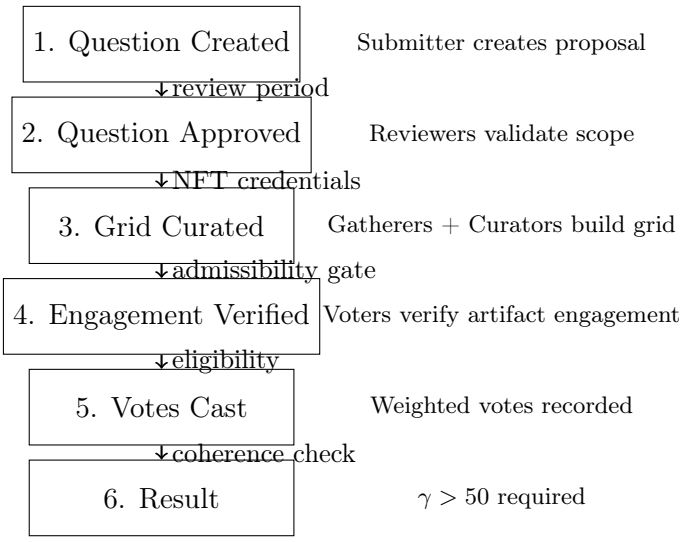


Figure 3: The Coherence Pipeline

Each stage has a gate. Failure at any gate halts progression:

Stage	Action	Gate	On Failure
1	Question created	Valid submitter NFT	Rejected
2	Question approved	Review period passes	Held for revision
3	Grid curated	Curator stake sufficient	Grid flagged
4	Engagement verified	Engagement threshold met	Voter ineligible
5	Vote cast	Signature valid, not duplicate	Vote rejected
6	Result	$\gamma > 50$ **and** majority	No state change

6.2 Stage 1: Question Creation

A registered `Submitter` creates a question (proposal):

```
// From lib.rs - submit_proposal
ensure!(
    matches!(participant.role, ParticipantRole::Submitter),
    Error::<T>::Unauthorized
);
```

The submitter must hold a valid credential NFT with the `Submitter` role. Questions enter the system in `Submitted` status.

Coherence dimension: Credential coherence—only credentialed participants can initiate state transitions.

6.3 Stage 2: Question Approval

Questions enter a mandatory review period:

```
voting_starts: current_block + T::ReviewPeriod::get(),
```

During review:

- Reviewers assess question clarity and scope
- Duplicate or malformed questions are flagged
- Community can signal concerns

Questions that survive the review period advance to `ReadyForVoting`.

Coherence dimension: Temporal coherence—deliberation has minimum dwell time.

6.4 Stage 3: Grid Curation

Parallel to review, credentialed **Gatherers** and **Curators** build the test grid:

1. **Gatherers** identify admissible artifacts: documents with valid provenance from recognized issuers (journals, agencies, standards bodies)

2. **Curators** assemble artifacts into engagement verification grids

3. Curators stake tokens on grid quality

4. Grid is published and linked to the proposal

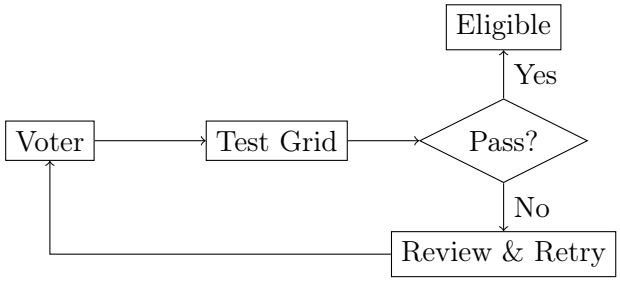
Curators hold `Curator` credential NFTs. Their stake is slashed if grids contain artifacts with invalid provenance, from non-recognized issuers, or that were retracted.

Important: Curators are *not* slashed for admitting artifacts whose conclusions are later contested or revised. The system does not adjudicate semantic disputes—only procedural validity.

Coherence dimension: Procedural coherence—the artifact registry is curated by accountable, credentialed participants with skin in the game.

6.5 Stage 4: Engagement Verification

Before voting, each participant must demonstrate engagement with proposal-relevant artifacts:



The engagement verification does not measure intelligence, political alignment, or agreement with artifact conclusions. It verifies that the voter has *encountered* the admissible artifacts relevant to this proposal.

Critical clarification: A voter who has engaged with the evidence and *disagrees* with its conclusions still passes the gate. The system verifies engagement, not agreement.

Passing the verification:

- Updates the voter's credential NFT (engagements verified counter)
- Grants eligibility to vote on this specific proposal
- Contributes to the voter's reputation score

Coherence dimension: Procedural coherence—voters demonstrate engagement with relevant artifacts before influencing outcomes.

6.6 Stage 5: Votes Cast

Eligible voters cast weighted votes:

```
// Weight calculation from lib.rs
let base_weight = participant.reputation;
let entropy_factor = (entropy[0] as u64 % 21) as i64 -
    10;
let quantum_adjustment = (base_weight as i64 *
    entropy_factor) / 100;
let final_weight = (base_weight as i64 +
    quantum_adjustment).max(1);
```

Vote weight is a function of:

1. **Reputation**: Accumulated from contributions, engagements verified, prior votes
2. **Quadratic cost**: Multiple votes on same proposal cost n^2
3. **Quantum entropy**: $\pm 10\%$ randomization prevents prediction

Each vote consumes quantum entropy and records a quantum signature. **Coherence dimensions**:

- Influence coherence (quadratic bounds)
- Process coherence (entropy randomization)

6.7 Stage 6: Result

After the voting period, finalization computes the result:

```
// The dual condition from lib.rs
let approved = approve_weight > reject_weight
    && quantum_confidence > 50;
```

Two conditions must **both** be satisfied:

1. **Democratic condition**: $w_{\text{approve}} > w_{\text{reject}}$
2. **Coherence condition**: $\gamma > 50$

The coherence score γ measures entropy variance across votes:

$$\gamma = \min\left(\frac{\sigma_\eta^2}{255}, 1\right) \times 100 \tag{5}$$

Low variance indicates correlated votes (Sybil attack, coordination). High variance indicates independent voting (legitimate process).

Critical clarification: The coherence score measures *process quality*, not outcome correctness. A proposal can pass with high coherence and still be "wrong" by external standards. The score measures whether the *process* exhibited properties consistent with legitimate deliberation, independent of proposal content.

Coherence dimension: Process coherence—the voting process itself must exhibit properties of statistical independence and adequate entropy.

6.8 The Integration Point

The pipeline stages are not independent. Each feeds the next:

$$\text{Credential} \xrightarrow{\text{enables}} \text{Engagement} \xrightarrow{\text{unlocks}} \text{Vote} \xrightarrow{\text{weighted by}} \text{Reputation} \xrightarrow{\text{checked by}} \text{Coherence} \tag{6}$$

Specifically:

- **Engagement pass → Reputation**: Each verified engagement increments reputation
- **Reputation → Vote weight**: Higher reputation = higher base weight
- **Vote weight → Coherence**: Weight distribution affects γ
- **Coherence → Credential**: Failed coherence doesn't update credentials

This creates a *feedback loop*: good-faith participation builds reputation, which increases influence, which is checked by coherence, which rewards legitimate behavior.

6.9 Why Both Admissibility and Process Coherence

One might ask: if voters demonstrate artifact engagement, why also check coherence?

The answer: **they protect against different attacks**.

Attack	**Blocked by Engagement**	**Blocked by γ**
Unengaged voting	✓	—
Sybil (fake identities)	Partial	✓
Coordinated manipulation	—	✓
Bribery	—	✓
Credential theft	—	✓
Bot voting	✓	✓

- **Admissibility gate** (engagement verification): Ensures voters have *encountered* the relevant artifacts
- **Coherence gate** (γ): Ensures votes are *statistically independent*

A voter who demonstrates engagement but coordinates with others will trigger low γ. A voter who votes independently but hasn't engaged with artifacts fails the admissibility gate. Both gates must pass for legitimate outcomes.

6.10 The Complete Coherence Invariant

Combining all dimensions, the system maintains a multi-dimensional coherence invariant:

Definition 6 (Democratic Coherence)**.** *A proposal outcome is* ***democratically coherent*** *if and only if:*

1. *All voters held valid credentials (credential coherence)*
2. *All voters demonstrated artifact engagement (procedural coherence)*
3. *Vote weights were bounded by quadratic costs (influence coherence)*
4. *Deliberation met minimum duration (temporal coherence)*
5. *Entropy distribution showed high variance (process coherence)*
6. *Majority of weighted votes approved (democratic condition)*

The system rejects outcomes that satisfy some but not all conditions. This is the "coherence preservation" of the central thesis:

> **Democratic legitimacy is a measurable property of process quality, independent of specific outcomes.**

Majority rule is necessary but not sufficient. The coherence invariant must also hold. The system constrains *how* decisions are made, not *what* decisions must conclude.

6.11 Pipeline Failure Modes

Each stage can fail, with different recovery paths:

Stage Failure	Symptom	Recovery
1. Invalid submitter	Rejected immediately	Get credential
2. Review rejection	Question held	Revise and resubmit
3. Grid challenge	Grid invalidated	New curators assigned
4. Engagement failure	Voter ineligible	Engage with artifacts, retry
5. Vote rejection	Vote not counted	Resubmit with valid sig
6. $\gamma \leq 50$	No state change	Investigate, revote

Critically: **no failure corrupts state**. The system remains at s_t until a coherent transition to s_{t+1} is achieved.

Important: Coherence failure ($\gamma \leq 50$) does not indicate that the proposal is wrong or harmful. It indicates only that the *process* exhibited statistical anomalies (coordination, manipulation, or entropy exhaustion).

6.12 Summary: One Pipeline, Six Coherence Checks

The augmented democracy system is a single pipeline:

$$\boxed{\text{Question} \rightarrow \text{Approval} \rightarrow \text{Grid} \rightarrow \text{Engagement} \rightarrow \text{Vote} \rightarrow \text{Result}}$$

Each transition is gated by a coherence check:

$$\boxed{\text{Credential} \rightarrow \text{Temporal} \rightarrow \text{Procedural} \rightarrow \text{Procedural} \rightarrow \text{Influence+Process} \rightarrow \text{Process}}$$

The "underlying current" is coherence preservation at every stage. The system is not a voting mechanism with add-ons—it is a coherence-preserving pipeline that happens to include voting as one stage.

This is the engineering realization of the central thesis: legitimacy emerges from process invariants, not outcome ratification.

The governance system constrains how decisions are made, not what decisions must conclude.

7 Coherence Constraints in Democratic Systems

7.1 Coherence Functionals for Proposals

Let $\mathcal{P}$ be the space of proposals and $\mathcal{C} : \mathcal{P} \rightarrow [0, 1]$ a coherence functional. A proposal p is *on-manifold* if $\mathcal{C}(p) > \tau$ for threshold τ.

In the implementation, $\mathcal{C}$ is realized through quantum confidence scoring:

$$\mathcal{C}(p) = \frac{\gamma_p}{100} \tag{7}$$

where γ_p is computed from entropy variance across votes on p.

Important: γ measures process quality, not outcome correctness. High γ indicates statistically independent participation with adequate entropy. Low γ indicates coordination, manipulation, or resource exhaustion.

7.2 Bounded Influence

No single participant can dominate outcomes. The weight function satisfies:

$$w_i \in [w_{\min}, w_{\max}] \quad \forall i \tag{8}$$

with $w_{\min} = 1$ (floor) and $w_{\max} = r_i \cdot 1.1$ (reputation cap with 10% entropy bonus).

7.3 Rollback and Retry Semantics

Failed coherence checks do not corrupt state. The system remains at s_t when:

- $w_{\text{approve}} \leq w_{\text{reject}}$ (democratic failure)
- $\gamma \leq 50$ (coherence failure)
- Entropy pool exhausted (resource failure)

Submitters may revise and resubmit. The state machine supports retry without penalty.

7.4 Limited Adversarial Feedback

The system limits adversarial feedback by restricting rejection signals to typed errors and coarse-grained status transitions. While no distributed system can eliminate all side-channel information, this design prevents gradient-style probing of coherence thresholds and minimizes adaptive manipulation.

Malicious proposals trigger explicit rejection with typed errors. The error taxonomy (Section 5) ensures that:

1. Attackers receive no fine-grained information about *why* rejection occurred
2. Legitimate users receive actionable feedback via different channels
3. All rejections are logged for audit

8 Quantum-Safe and Post-Quantum Alignment

This is not a cryptography paper. However, the governance system inherits quantum resistance from its substrate:

8.1 Why Classical Assumptions Fail

- RSA/ECDSA signatures are broken by Shor's algorithm
- Hash-based commitments remain secure (Grover provides only quadratic speedup)
- Retroactive compromise: adversaries may record today's traffic for future decryption

8.2 Post-Quantum Protections

The Quantum Harmony implementation uses:

- **SPHINCS+-256s**: Stateless hash-based signatures (NIST PQC standard)
- **Blake2-256**: Quantum-resistant hashing
- **QKD entropy**: True quantum randomness from KIRQ network

8.3 Proof-of-Coherence vs. Proof-of-Work/Stake

Property	**PoW**	**PoS**	**PoC**
Energy cost	High	Low	Low
Plutocracy risk	Medium	High	Low
Quantum resistance	None	None	Full
Sybil resistance	Hash rate	Stake	Coherence
Manipulation detection	None	None	Built-in

Proof-of-Coherence detects manipulation *before* state transition, whereas PoW/PoS detect only after finalization (if at all).

9 Historical Implementation: From Concept to Quantum Infrastructure

The augmented democracy framework has evolved through eight years of development, from theoretical conception to production deployment.

9.1 Conceptual Foundation (2017)

The original framework was published in August 2017 as "A Machine-Based Societal Model for Curbing Citizen Cynicism" [1]. Key concepts established:

- **Test grids**: Voters must demonstrate engagement with relevant evidence

- **Mechanical humans**: Low-barrier contribution tasks for universal participation
- **Operating system metaphor**: Governance as a self-regulating system
- **Blockchain validation**: Immutable record of all decisions
- **Official as safeguard**: Human oversight of machine recommendations

The 2017 paper explicitly framed democracy as "critical infrastructure" requiring engineering discipline—a perspective that remains central to the current implementation.

9.2 EOS Prototype and Oracle Integration (2020–2021)

The first production implementation deployed as an EOS smart contract:

- **augdemocracy.cpp**: Core smart contract with registration and voting
- **Token Curated Test Grids**: KILT-inspired gatherer/curator structure [8]
- **Oracle integration**: Decentralized artifact sourcing via oracle networks
- **Dynamic NFTs**: Early credential management prototypes
- **Quadratic voting**: Whale resistance mechanisms

The EOS implementation validated the admissibility gate concept in production, demonstrating that artifact-engagement verification was technically feasible and user-acceptable. Concurrent research into Polkadot ecosystem oracle solutions (including Kylin Network, a Polkadot parachain for cross-chain data feeds) informed the design of the decentralized artifact-sourcing layer.

9.3 Substrate Migration (2022–2023)

Migration to the Substrate framework enabled:

- **Pallet architecture**: Modular, upgradeable governance components
- **Cross-chain compatibility**: Polkadot ecosystem integration
- **Runtime upgrades**: Governance system self-modification
- **Improved NFT standards**: FRAME-native credential tokens

The Substrate pallet (`pallet-quantum-democracy`) preserved all EOS-era features while adding formal verification capabilities.

9.4 Quantum Harmony Integration (2024–2025)

Current production deployment on Quantum Harmony blockchain:

- **Quantum entropy pool**: True randomness from KIRQ network
- **Coherence-based consensus**: Replaces stake-based finality
- **Post-quantum signatures**: SPHINCS+-256s for credential integrity
- **QKD-protected channels**: Quantum key distribution for artifact delivery
- **Toshiba/Crypto4A integration**: Hardware security module support

9.5 Continuity of Core Principles

Across all implementations, core principles remain constant:

Principle	2017	EOS	Substrate	QH
Artifact-gated voting	Concept	Implemented	Enhanced	Production
Dynamic credentials	Concept	Prototype	NFT-based	Life-sustaining
Quadratic bounds	Concept	Implemented	Integrated	+ Entropy
Coherence threshold	Implicit	Partial	Formal	Quantum
Official safeguard	Central	Preserved	Root-only	Emergency

This evolution demonstrates that augmented democracy is not speculative—it is iteratively refined infrastructure with continuous production deployment since 2020.

10 Failure Modes and Emergency Governance

10.1 When Coherence Fails

Coherence failure ($\gamma \leq 50$) indicates:

1. **Coordinated attack**: Sybil voters with correlated entropy
2. **Entropy exhaustion**: Insufficient quantum randomness
3. **System compromise**: Entropy source manipulation

Coherence failure does *not* indicate that the proposal is wrong, harmful, or should be rejected on substantive grounds. It indicates only that the *process* exhibited statistical anomalies.

10.2 Emergency Committee Override

For critical failures, the system supports:

```
ensure_root(origin)?;  // Root-only operations
```

This is the "break glass" mechanism. Root authority can:

- Replenish entropy pool
- Pause voting
- Trigger manual review

Root cannot:

- Directly approve/reject proposals
- Modify vote records
- Alter coherence thresholds retroactively

10.3 Human-in-the-Loop Escalation

When automated coherence checks fail repeatedly, the system escalates to human review:

1. Proposal flagged for manual assessment
2. Validator committee convenes (off-chain)
3. Committee issues signed recommendation
4. Recommendation enters normal voting pipeline

This preserves the coherence constraint while acknowledging that some edge cases require human judgment.

10.4 Response Windows

Event	Detection	Response
Low entropy	Immediate	Auto-pause voting
Coherence failure	At finalization	Proposal held
Repeated failures	3 consecutive	Escalate to committee
System compromise	Manual detection	Emergency halt

11 User Experience: Hiding the Machinery

The preceding sections describe protocol mechanics. This section addresses the critical question: *how does a citizen actually use this?*

The coherence pipeline has six stages and multiple credential types. Exposing this complexity to end users would guarantee adoption failure. The system must feel like "just voting" while the coherence machinery runs invisibly beneath.

11.1 The TCP/IP Principle

Users do not understand TCP/IP, TLS handshakes, or DNS resolution. They "browse the web." The complexity exists but is entirely hidden by the browser interface.

Augmented democracy requires the same architectural separation:

Layer	**What User Sees**
Protocol	(invisible)
Application	Simple voting interface
Experience	"I voted on the transit proposal"

The test grids, NFT credentials, quadratic costs, and coherence thresholds operate at the protocol layer. The user interacts with the application layer.

11.2 The Streamlined Experience

For a typical citizen voting on a local issue:

1. **Open app, see proposals**: "New transit line proposal"
2. **Tap to learn more**: 2-minute summary video + key information
3. **Quick quiz appears**: 3 questions, multiple choice
4. **Pass quiz**: "You're ready to vote!"
5. **Vote**: Approve / Reject / Abstain
6. **Done**: "Your vote is recorded"

Total time: **4 minutes**.
What happened invisibly:

- App checked credential NFT validity
- Quiz was the engagement verification grid (presented as "quick quiz")

- Pass threshold was applied
- Vote was signed with quantum signature
- Quadratic cost was calculated (first vote = 1 token, shown as "free")
- Vote entered coherence pool

The user experienced: "watched video, answered quiz, voted."

11.3 Progressive Disclosure

Different users need different depth:

User Type	Sees	Hidden
Casual voter	Quiz + vote button	Everything else
Engaged citizen	Reputation score, vote history	NFT mechanics
Power user	Credential details, weight calc	Protocol internals
Curator	Grid builder interface	Consensus mechanics
Developer	Full protocol access	Nothing

The interface progressively reveals complexity only when users seek it.

11.4 The Quiz Is the Engagement Verification

The "test grid" sounds bureaucratic. The experience is:

> "Before you vote, let's make sure you've seen the key information. Here are 3 quick questions."

The questions verify engagement with the evidence:

- "The proposed transit line would cost approximately: (a) $2B (b) $5B (c) $10B"
- "The project timeline is: (a) 2 years (b) 5 years (c) 10 years"
- "The main opposition concern is: (a) cost (b) displacement (c) noise"

Critical clarification: This is not an IQ test or agreement test. It verifies the voter has *encountered* the relevant information. A voter who has engaged with the evidence and *disagrees* with it still passes. The system verifies engagement, not agreement.

Failing means: "Review the summary and try again" — not punishment, just a nudge to engage with the material.

11.5 Credential Management Is Invisible

Users never see "NFT credential" language. They see:

- **Account creation**: "Sign up with email" (NFT minted in background)
- **Reputation**: "Your civic score: 127" (reputation field)
- **Eligibility**: "You can vote on 12 active proposals" (credential check)
- **Decay warning**: "Vote this month to keep your streak!" (vitality)

The gamification layer (scores, streaks, badges) maps directly to protocol primitives (reputation, vitality, domain credentials) without exposing the underlying mechanics.

11.6 Quadratic Costs as "Voting Power"

Quadratic voting sounds academic. The UX:

> "You have 100 voting power this month."
> "Spending 1 power = 1 vote."
> "Spending 4 power = 2 votes (for issues you care deeply about)."
> "Spending 9 power = 3 votes."

Users understand "spend more to vote stronger on things you care about." The quadratic cost is implicit in the power/vote ratio.

Most users spend 1 power per proposal and never think about the math.

11.7 The Curator Path

For users who want deeper engagement, the curator path opens:

1. **Contribute sources**: "Add a source document to this proposal"
2. **Build quizzes**: "Write a question for this proposal"
3. **Earn tokens**: "You earned 5 tokens for your contribution"
4. **Gain reputation**: "Your civic score increased to 142"

Curators experience the system as a contribution platform with rewards. The staking, slashing, and credential mechanics are protocol-level concerns.

11.8 Failure States as Friendly Nudges

Protocol failures translate to friendly UX:

Protocol State	User Message
Engagement not verified	"Review the summary and try the quiz again"
Credential expired	"Welcome back! Quick verification needed"
Insufficient power	"You've used your voting power this month"
Coherence failure	"This vote is under review—we'll notify you"

No user ever sees "InsufficientQuantumEntropy" or "CoherenceThresholdNotMet."

11.9 The 2017 Vision: Comfortable Contribution

The original 2017 paper emphasized that contribution must be "comfortable":

> "An individual must be granted the freedom to choose their level of involvement, the type of involvement they want to partake in, and the level of privacy they want for a particular transaction."

This translates to:

- **Level of involvement**: Casual voter vs. curator vs. developer
- **Type of involvement**: Vote, contribute sources, build quizzes, write code
- **Privacy level**: Public reputation vs. anonymous mode

The system accommodates all engagement levels without forcing complexity on those who don't want it.

11.10 Mobile-First, 4-Minute Sessions

The target interaction:

- **Platform**: Mobile app (iOS/Android)
- **Session length**: 2–5 minutes
- **Notification**: "3 proposals need your vote this week"
- **Interaction**: Watch summary → quiz → vote
- **Reward**: "+5 civic score, streak maintained"

This is the friction target: **less friction than checking social media**.

11.11 Incentive Alignment

Why would anyone participate?

Incentive	Mechanism
Civic score	Social status, visible to others
Streaks	Gamification, loss aversion
Token rewards	Direct compensation for curation
Influence	Higher reputation = more vote weight
Tax benefits	Civic contribution deductions (policy)

The 2017 paper proposed that civic contribution could offset taxes and unlock social benefits. The exact incentive structure is policy, not protocol—but the protocol supports any incentive model.

11.12 Summary: The Experience Stack

Layer	Contents
Experience	"I voted on transit"
Interface	Quiz, vote button, score display
Application	Credential check, weight calc, submission
Protocol	NFTs, entropy, signatures, coherence
Consensus	Quantum Harmony blockchain

The protocol is complex. The experience is simple.

This is the same pattern as the modern web: users don't understand HTTPS, but they trust the lock icon. Users won't understand coherence thresholds, but they'll trust "Your vote is verified."

The machinery serves the experience. Not the reverse.

11.13 Agent-Mediated Participation

The preceding subsections assume human users navigating a simplified interface. A more radical approach: **delegate to AI agents**.

Humans don't navigate TCP/IP because *browsers* do it for them. Similarly, humans may not navigate the coherence pipeline because *agents* do it for them.

11.13.1 What Agents Can Do

An AI agent operating on behalf of a citizen can:

- **Monitor proposals**: "3 new proposals match your interests"

- **Summarize content**: Digest 50-page proposals into 2-minute briefings
- **Complete engagement verifications**: Parse test grids and demonstrate artifact engagement
- **Maintain credentials**: Ensure NFT vitality, renew before decay
- **Calculate spending**: Optimize quadratic voting across proposals
- **Vote within bounds**: Execute votes per user-defined preferences
- **Track coherence**: Alert user if coherence scores are anomalous

The human experience becomes:

1. **Set preferences**: "I care about transit, housing, environment"
2. **Review agent recommendations**: "Your agent suggests: Approve transit, Reject rezoning"
3. **Confirm or override**: Tap to accept, or dive deeper
4. **Done**: Agent handles everything else

Total human time: **30 seconds per week**.

11.13.2 Delegation Bounds

Agents operate within *delegation bounds* set by the user:

```rust
pub struct DelegationBounds {
    // What the agent CAN do autonomously
    pub can_complete_engagements: bool,
    pub can_vote_low_stakes: bool,  // < threshold
    pub can_manage_credentials: bool,

    // What requires human confirmation
    pub vote_threshold: Balance,    // Above this, ask human
    pub domains_excluded: Vec<Domain>, // Never vote on these
    pub require_confirmation: bool, // Always ask before voting

    // Limits
    pub max_daily_votes: u32,
    pub max_weekly_spend: Balance,
}
```

A conservative user might set: "Agent can complete engagement verifications, but always ask me before voting."

A trusting user might set: "Agent can vote on anything under 10 tokens; ask me for high-stakes only."

11.13.3 Agent Accountability

Critical question: *who is accountable for agent actions?*

The answer: **the human principal**.

- Agent actions are signed with the human's credential
- Vote history shows "voted via agent" flag
- Reputation changes accrue to the human
- The human remains accountable for agent behavior

The agent is a tool, not a participant. The human delegates but remains accountable.

11.13.4 Agent Coherence (ERLHS Connection)

If agents vote, the agents themselves need coherence constraints.

This connects directly to ERLHS [2]: the Hamiltonian framework for coherence-preserving machine intelligence. An agent operating in the augmented democracy system should:

- Maintain internal state coherence (ERLHS constraint)
- Respect delegation bounds (policy constraint)
- Produce auditable reasoning (transparency constraint)
- Avoid manipulation of other agents (adversarial constraint)

The coherence requirements apply at two levels:

1. **Democratic coherence**: The voting system (this paper)
2. **Agent coherence**: The AI operating within it (ERLHS)

11.13.5 Mass Agent Attacks

New threat model: adversary deploys many agents to vote in coordinated patterns.

Defenses:

- **Coherence detection**: Coordinated agents produce low entropy variance (γ)
- **Agent diversity requirements**: Agents must demonstrate behavioral variance
- **Human-in-loop checkpoints**: High-stakes votes require human confirmation
- **Agent registration**: Agents themselves may need credentials

The quantum confidence score γ was designed to detect Sybil attacks. It also detects coordinated agent behavior—agents voting in lockstep produce low variance, triggering coherence failure.

11.13.6 The Agent-Native Interface

For agent-mediated participation, the "interface" is an API:

```
trait AgentInterface {
    fn get_proposals(&self, filters: ProposalFilters)
        -> Vec<ProposalSummary>;
    fn get_engagement_grid(&self, proposal_id: u32)
        -> EngagementGrid;
    fn submit_engagement_answers(&self, proposal_id: u32,
        answers: Vec<Answer>)
        -> EngagementResult;
    fn cast_vote(&self, proposal_id: u32, vote: VoteType,
        weight: u64)
        -> VoteReceipt;
    fn get_coherence_status(&self, proposal_id: u32)
        -> CoherenceStatus;
}
```

Human-facing apps and agent systems use the same underlying protocol. The difference is who drives the interaction.

11.13.7 The Future: Humans Set Values, Agents Execute

The end state:

Human Role	Agent Role
Set values and priorities	Monitor proposal stream
Review agent recommendations	Summarize and analyze
Confirm high-stakes votes	Execute routine votes
Override when needed	Maintain credentials
Remain accountable	Optimize participation

This is not "AI voting instead of humans." It is "AI handling friction so humans can focus on values."

The human remains the principal. The agent is the instrument. The coherence constraints apply to both.

The governance system constrains how decisions are made, not what decisions must conclude.

12 Conclusion: Democracy as Infrastructure

Democracy is no longer a philosophical abstraction. It is **critical infrastructure**.

The systems we use to make collective decisions are under adversarial load from state actors, AI-generated influence, and the fundamental speed/quality tradeoff of digital communication. Traditional democratic theory offers no defense against these threats because it treats legitimacy as a property of outcomes rather than processes.

This paper has presented an alternative: *coherence-constrained democratic systems* with *procedural infrastructure*, where:

1. Proposals are state transitions in a formal control system
2. Participants hold dynamic credentials that evolve with contribution
3. Voters must demonstrate engagement with admissible artifacts (procedural gate)
4. Quadratic voting costs bound plutocratic influence
5. Deliberation is a filtering operation with minimum dwell time
6. Approval requires both majority support *and* coherence threshold
7. All transitions produce cryptographic proofs
8. Credentials decay without ongoing participation (life-sustaining NFTs)

The central thesis is that **democratic legitimacy is a measurable property of process quality, independent of specific outcomes**.

This is not ideology. It is engineering.

The governance system constrains how decisions are made, not what decisions must conclude.

The framework has evolved through eight years of development—from the 2017 conceptual paper through EOS smart contract prototypes to the current Quantum Harmony implementation. Each iteration validated core principles while adding cryptographic hardening. Combined with prior work on ERLHS (coherence in AI), Karmonic Mesh (geometric substrate), and Proof of Coherence (distributed validation), this framework provides a complete architecture for augmented democracy as 21st-century infrastructure.

The procedural stack—token-curated test grids, dynamic credential NFTs, quadratic voting, coherence thresholds, and accountability mechanisms—transforms voting from opinion aggregation into *structured preference revelation*. This is the augmentation that gives augmented democracy its name: not technological enhancement of voting mechanics, but procedural elevation of participation quality.

The friction inherent in this architecture is addressed through agent-mediated participation: AI agents that navigate the coherence pipeline on behalf of human principals. Humans set values and priorities; agents handle the mechanics. This creates a new division of labor—humans as value-setters, agents as executors— with coherence constraints applying at both levels. The ERLHS framework ensures agent coherence; the democratic framework ensures process coherence. Together, they enable participation at scale without sacrificing procedural quality.

The ultimate vision: a system where legitimate collective decisions emerge from coherence-preserving processes, mediated by trustworthy agents, grounded in verifiable artifacts, and accountable to human values. This is democracy as infrastructure— engineered, deployable, and resistant to adversarial conditions.

References

[1] Cormier, S. (2017). *A Machine-Based Societal Model for Curbing Citizen Cynicism*. Unpublished manuscript, August 2017.

[2] Cormier, S. (2025). *ERLHS: A Hamiltonian Framework for Coherence-Preserving Machine Intelligence*. Zenodo. `https://doi.org/10.5281/zenodo.17928909`

[3] Cormier, S. (2025). *Karmonic Mesh: Spectral Consensus on Toroidal Manifolds*. Zenodo. `https://doi.org/10.5281/zenodo.17928991`

[4] Cormier, S. (2025). *Proof of Coherence: QKD-Based Distributed Consensus*. Zenodo. `https://doi.org/10.5281/zenodo.17929054`

[5] Cormier, S. (2025). *Toroidal Mesh: 10K TPS with SPHINCS+ via Parallel Verification*. Zenodo. https://doi.org/10.5281/zenodo.17931222

[6] Cormier, S. (2025). *Toroidal Governance: Tonnetz Manifold Democratic Infrastructure*. Zenodo. https://doi.org/10.5281/zenodo.17929091

[7] Buterin, V., Hitzig, Z., & Weyl, E. G. (2019). *A Flexible Design for Funding Public Goods*. Management Science, 65(11), 5171–5187.

[8] BOTLabs GmbH. (2020). *KILT Protocol White Paper: Credentials for Web 3.0*. Berlin, Germany. https://kilt-protocol.org/files/KILT-White-Paper.pdf